

CUBES-SIL Clumped CO2 data processing

code by Katie Snell, updated 06/15/2020

To do list:

- Should we be using york regressions for all the lines? E.g., including std correction lines, ARF ETFs - probably
 - have a test version in there (actual ARF values are still based on the linear model), but I'm not sure what to do about the fact that we don't have errors for the y values. Can't run it with "0" as errors, and with very small errors, is effectively the same as the standard linear model
- Make error contours for D47 line, like Mathieu's web app?
- correct d13C and d18O values

Data processing Scheme

1. Read data and cull samples with known analytical problems
2. Determine D48 excess and label samples
 - Calculates "true" values relative to HG, then assigns "clean", "maybe", and "dirty" based on theoretical separate between HG line and EG line of 0.4
3. Make heated gas D47 line from data with no D48 excess
4. Calculate external D47 errors (includes error from heated gas line)
5. Initial data corrections: HG line, stretching, internal acid digestion correction
6. Absolute Reference Frame
 - 6.1. Make ETF lines from data with no D48 excess and calculate ARF D47 values
 - 6.1.1: Primary Approach - uses heated and equilibrated gases
 - 6.1.2: Secondary Approach - uses standards
 - 6.2. Calculate standard residuals from ARF D47 values
 - 6.3. Standard corrections for ARF D47 values from data with no D48 excess using a line between the standards
 - 6.4. Calculate temperatures from acid digestion corrected values, and all std corrected columns

Data processing code

1. Load data, packages and cull samples with known analytical problems

MANUALLY UPDATE: input your file name and give the session a name

```
User <- "Marina"
session <- "DH3" #change this to whatever name or number you want to give you session
MS.ID <- "Bert"
data <- read.csv("DH2_flatlist.csv")
data$MS <- MS.ID
```

load necessary R libraries and clumped isotope standards data

```
#devtools::install_github("cubessil/CUBESSILclumpedmath") #general CUBESsil lab stuff
#devtools::install_github("cubessil/clumpsDRtools") #only for processing clumped data

library(plotly)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
## last_plot
```

```
## The following object is masked from 'package:stats':
##
## filter
```

```
## The following object is masked from 'package:graphics':
##
## layout
```

```
library(shiny)
library(isoprocessCUBES)
library(tidyverse)
```

```
## — Attaching packages —
## tidyverse 1.3.0 —
```

```
## ✓ tibble 3.0.1 ✓ dplyr 0.8.5
## ✓ tidyr 1.0.2 ✓ stringr 1.4.0
## ✓ readr 1.3.1 ✓ forcats 0.5.0
## ✓ purrr 0.3.4
```

```
## — Conflicts —
## tidyverse_conflicts() —
## x dplyr::filter() masks plotly::filter(), stats::filter()
## x dplyr::lag() masks stats::lag()
```

```
library(openxlsx)
library(CUBESSILclumpedmath)
```

```
##
## Attaching package: 'CUBESSILclumpedmath'
```

```
## The following objects are masked from 'package:isoprocessCUBES':
##
## calc_means, york.regression
```

```
#library(clumpsDRtools)
```

```
Standards <- readRDS(url("https://github.com/cubessil/clumpsDRcodes/blob/master/Standards.RDS?raw=true"))
```

Fixing name mistakes, subset data to project, drop remaining unrelated samples

```
data <- filter(data, Project != "NA")
```

Calculate internal standard errors of the mean for each column of isotope data, using number of acq for each sample

```
data$d48.sterr<-round(data$d48.stdev/sqrt(data$num.acq), 4)
data$d48.sterr<-round(data$d48.stdev/sqrt(data$num.acq), 4)
data$d47.sterr<-round(data$d47.stdev/sqrt(data$num.acq), 4)
data$d47.sterr<-round(data$d47.stdev/sqrt(data$num.acq), 4)
data$d13C.sterr<-round(data$d13C.stdev/sqrt(data$num.acq), 4)
data$d13C.sterr<-round(data$d13C.stdev/sqrt(data$num.acq), 4)
data$d180.sterr<-round(data$d180.stdev/sqrt(data$num.acq), 4)
data$d180.sterr<-round(data$d180.stdev/sqrt(data$num.acq), 4)
data$d45.sterr<-round(data$d45.stdev/sqrt(data$num.acq), 4)
data$d46.sterr<-round(data$d46.stdev/sqrt(data$num.acq), 4)
```

2. Determine D48 excess and label samples

Use all HQ and EQ gas data to determine D48 excess

* Optional step **not applied**: repeat D48 excess determination after D48 excess removed

Define D48 HG dataframe for 48 line (exclude samples with analytical problems, run initial york regression, use results to calculate more realistic errors, then perform final york regression)

```
HG48data <- data %>%
  filter(Donotuse != TRUE & Type=="heated gas") %>%
  select(Sample.ID, num.acq, Donotuse, spec.num, d48, D48, D48.sterr, d48.sterr)

D48line.initial <- cbind(data.frame(Session=session, MS=MS.ID, line.ID="initial D48line"), york.regression(HG48data$d48, HG48data$d48.sterr, HG48data$d48, HG48data$d48.sterr))

HG48data<-calc_exterr(HG48data, 48, D48line.initial$york.slope, D48line.initial$york.intercept, D48line.initial$N)

D48line.final <- cbind(data.frame(Session=session, MS=MS.ID, line.ID="final D48line"), york.regression(HG48data$d48, HG48data$d48.sterr, HG48data$d48, HG48data$d48.sterr))

D48lines<-rbind(D48line.initial, D48line.final)
print(D48lines)
```

```
## Session MS line.ID york.slope york.slope.err york.intercept
## 1 DH3 Bert initial D48line 0.07188459 0.0002386609 -0.3819867
## 2 DH3 Bert final D48line 0.07180920 0.0005929465 -0.3757578
## york.int.err MSWD N Cor.ab error.corr
## 1 0.007387157 20.52083 28 9.174859e-07 0.9845513
## 2 0.018463839 3.44779 28 5.772863e-06 0.9856182
```

Calculate D48 HG corrected values using line and predictive and confidence intervals for D48 regression (note D48 excess values = D48 HG corrected values IF the 48 line used to define excess is only made up of HGs). Also, use this approach when there is no equilibrated gas to define the upper bound; here we use theoretical bound of 0.4 as the upper EG value.

```
D48data <- data %>%
  select(Sample.ID, num.acq, spec.num, Type, Donotuse, d48, D48, D48.sterr, d48.sterr)

D48data<-york.conf.pred.interval(HG48data, D48data, 48, D48line.final$york.slope, D48line.final$york.intercept, 0.95)

D48data$d48.HGcorr <- D48data$d48 - (D48data$d48 * D48line.final$york.slope + D48line.final$york.intercept)

D48data$d48excess.label <= "no D48 excess"
```

```
## logical(0)
```

```
D48data <- D48data %>%
  mutate(d48excess.label = ifelse(D48.HGcorr >= 0.345 & D48.HGcorr <= 0.345 + D48.pred.int, "maybe D48 excess",
    ifelse(D48.HGcorr > 0.345 + D48.pred.int, "D48 excess", "no D48 excess")))

data <- full_join(data, select(D48data, spec.num, D48.HGcorr, D48excess.label), by = "spec.num")
```

Plot and save heated gas 48 line with confidence and predictive intervals

```

# colored by 48 excess, shaped by type
D48.excess1_time<-ggplot(filter(D48data, Donotuse !=TRUE), aes(x=d48, y=D48)) +
  geom_ribbon(aes(ymin=all.predicted.Y-D48.pred.int, ymax=all.predicted.Y+D48.pred.int), fill="gray") +
  geom_ribbon(aes(ymin=all.predicted.Y-D48.conf.int, ymax=all.predicted.Y+D48.conf.int), fill="white") +
  geom_abline(data=D48line.final, aes(slope=york.slope, intercept=york.intercept), size=0.5) +
  geom_point(aes(colour=D48excess.label), size=3) +
  geom_point(aes(shape=Type, fill=D48excess.label), size=3.5) +
  scale_shape_manual(values=c(21,22,24,25,23)) +
  annotate("text", x = -20, y = 29, label = york_eqn(D48line.final$york.slope, D48line.final$york.intercept), siz
e = 4, hjust=0, vjust=0, parse=TRUE) +
  guides(fill=FALSE) +
  theme_bw() +
  labs(title= paste0(session, " D48 line"))

# colored by type, shaped by 48 excess
D48.excess2_time<-ggplot(filter(D48data, Donotuse !=TRUE), aes(x=d48, y=D48)) +
  geom_ribbon(aes(ymin=all.predicted.Y-D48.pred.int, ymax=all.predicted.Y+D48.pred.int), fill="gray") +
  geom_ribbon(aes(ymin=all.predicted.Y-D48.conf.int, ymax=all.predicted.Y+D48.conf.int), fill="white") +
  geom_abline(data=D48line.final, aes(slope=york.slope, intercept=york.intercept), size=0.5) +
  geom_point(aes(colour=Type), size=3) +
  geom_point(aes(fill=Type, shape=D48excess.label), size=3.5) +
  scale_shape_manual(values=c(21,22,24,25,23)) +
  annotate("text", x = -20, y = 29, label = york_eqn(D48line.final$york.slope, D48line.final$york.intercept), si
ze = 4, hjust=0, vjust=0, parse=TRUE) +
  guides(fill=FALSE) +
  theme_bw() +
  labs(title= paste0(session, " all data D48 line, by type"))

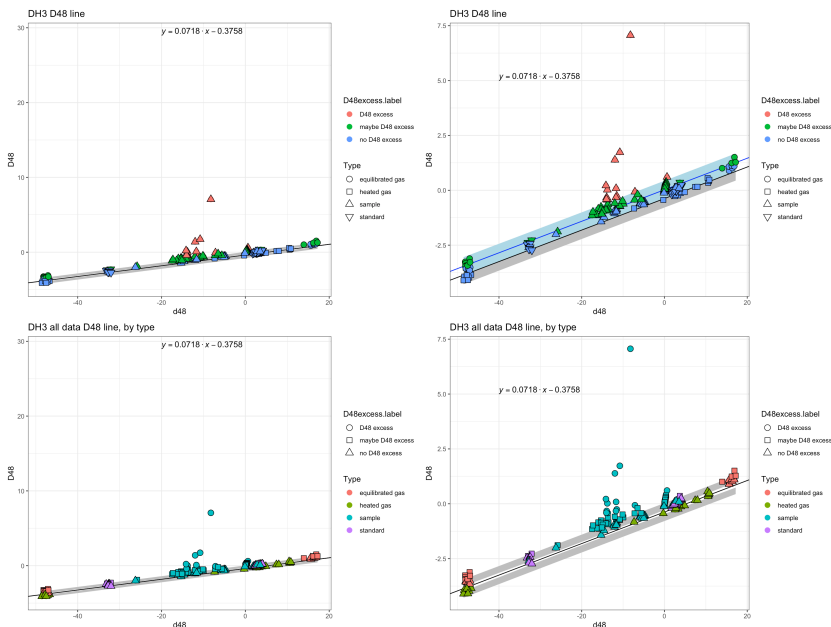
D48.excess3_time<-ggplot(filter(D48data, Donotuse !=TRUE), aes(x=d48, y=D48)) +
  geom_ribbon(aes(ymin=all.predicted.Y-D48.pred.int, ymax=all.predicted.Y+D48.pred.int), fill="gray") +
  geom_ribbon(aes(ymin=0.4+all.predicted.Y-D48.pred.int, ymax=0.4+all.predicted.Y+D48.pred.int), fill="light blu
e") +
  geom_abline(data=D48line.final, aes(slope=york.slope, intercept=york.intercept), size=0.5) +
  geom_abline(data=D48line.final, aes(slope=york.slope, intercept=0.4+york.intercept), size=0.5, color = "blue")
+
  geom_point(aes(colour=D48excess.label), size=3) +
  geom_point(aes(shape=Type, fill=D48excess.label), size=3.5) +
  scale_shape_manual(values=c(21,22,24,25,23)) +
  annotate("text", x = -40, y = 5, label = york_eqn(D48line.final$york.slope, D48line.final$york.intercept), size
= 4, hjust=0, vjust=0, parse=TRUE) +
  guides(fill=FALSE) +
  theme_bw() +
  labs(title= paste0(session, " D48 line"))

# colored by type, shaped by 48 excess
D48.excess4_time<-ggplot(filter(D48data, Donotuse !=TRUE), aes(x=d48, y=D48)) +
  geom_ribbon(aes(ymin=all.predicted.Y-D48.pred.int, ymax=all.predicted.Y+D48.pred.int), fill="gray") +
  geom_ribbon(aes(ymin=all.predicted.Y-D48.conf.int, ymax=all.predicted.Y+D48.conf.int), fill="white") +
  geom_abline(data=D48line.final, aes(slope=york.slope, intercept=york.intercept), size=0.5) +
  geom_point(aes(colour=Type), size=3) +
  geom_point(aes(fill=Type, shape=D48excess.label), size=3.5) +
  scale_shape_manual(values=c(21,22,24,25,23)) +
  annotate("text", x = -40, y = 5, label = york_eqn(D48line.final$york.slope, D48line.final$york.intercept), siz
e = 4, hjust=0, vjust=0, parse=TRUE) +
  guides(fill=FALSE) +
  theme_bw() +
  labs(title= paste0(session, " all data D48 line, by type"))

multiplot(D48.excess1_time, D48.excess2_time, D48.excess3_time, D48.excess4_time, cols=2)

```

```
## Loading required package: grid
```



2.b Create CDES reference frame values for D48 HG corrected values

```

CDES48line.pri <- data %>%
  filter(D48excess.label=="no D48 excess")%>%
  filter(Donotuse != TRUE & Type=="heated gas" | Type=="equilibrated gas") %>%
  left_join(select(Standards, Sample.ID, D48.CDES.90.acc), by = "Sample.ID")

```

```

## Warning: Column `Sample.ID` joining factors with different levels, coercing to
## character vector

```

```

CDES.48slope.pri <- (coef(lm(CDES48line.pri$D48.CDES.90.acc ~ CDES48line.pri$D48.HGcorr)))[2]]
CDES.48intercept.pri <- (coef(lm(CDES48line.pri$D48.CDES.90.acc ~ CDES48line.pri$D48.HGcorr)))[1]]

data$CDES.48noAD.pri <- CDES.48slope.pri*data$D48.HGcorr+CDES.48intercept.pri #calculates pre-Acid digestion corrected CDES values

#calculating version based on means:
CDES48line.pri.mean <- data %>%
  filter(D48excess.label=="no D48 excess")%>%
  filter(Donotuse != TRUE & Type=="heated gas" | Type=="equilibrated gas") %>%
  left_join(select(Standards, Sample.ID, D48.CDES.90.acc), by = "Sample.ID") %>%
  group_by(Sample.ID) %>%
  summarize(D48.accepted = mean(D48.CDES.90.acc),
            D48.measured = mean(D48.HGcorr))

```

```

## Warning: Column 'Sample.ID' joining factors with different levels, coercing to
## character vector

```

```

CDES.48slope.pri.mean <- (coef(lm(CDES48line.pri.mean$D48.accepted ~ CDES48line.pri.mean$D48.measured)))[2]]
CDES.48intercept.pri.mean <- (coef(lm(CDES48line.pri.mean$D48.accepted ~ CDES48line.pri.mean$D48.measured)))[1]]

data$CDES.48noAD.pri <- CDES.48slope.pri.mean*data$D48.HGcorr+CDES.48intercept.pri.mean #calculates pre-Acid digestion corrected CDES values
data$CDES.48noAD.pri.all <- CDES.48slope.pri*data$D48.HGcorr+CDES.48intercept.pri #calculates pre-Acid digestion corrected CDES values

```

Plots CDES D48 transfer function line from just the reference gas data - the regression in this plot is based only on the heated and equilibrated gas data. The larger standard symbols show the internal heated gas corrected values versus the ARF values they have using the results of this regression, and the smaller sample values show the internal Acid digestion corrected values versus the ARF acid digestion corrected values.

CHECK: if there are no problems with the acid digestion only, AD-corrected values should fall on the line; IS THIS REALLY TRUE?

*Note that in this plot, the line is regressed based on ONLY the gas data. The big standards are the pre-acid digestion corrected values, and the small standards are the acid digestion corrected values....

```

pCDES48line.pri <- ggplot(CDES48line.pri, aes(x=D48.HGcorr, y=D48.CDES.90.acc)) +
  geom_point(aes(fill=Sample.ID, shape=Sample.ID), size=4) +
  stat_smooth(method="lm") +
  geom_abline(slope = CDES.48slope.pri.mean, intercept = CDES.48intercept.pri.mean, colour = "red") +
  geom_point(data=subset(data, Type=="standard" & D48excess.label=="no D48 excess"), aes(x=D48.HGcorr, y=CDES.48noAD.pri, fill=Sample.ID, shape=Sample.ID, show_guide=FALSE), size=3) +
  scale_shape_manual(values=c(21,22,23,24,25,21,22,23,24)) +
  labs(title=paste0(session, " Primary CDES transfer function"), x="Internal D48", y="CDES D48") +
  theme_bw()

```

```

## Warning: Ignoring unknown aesthetics: show_guide

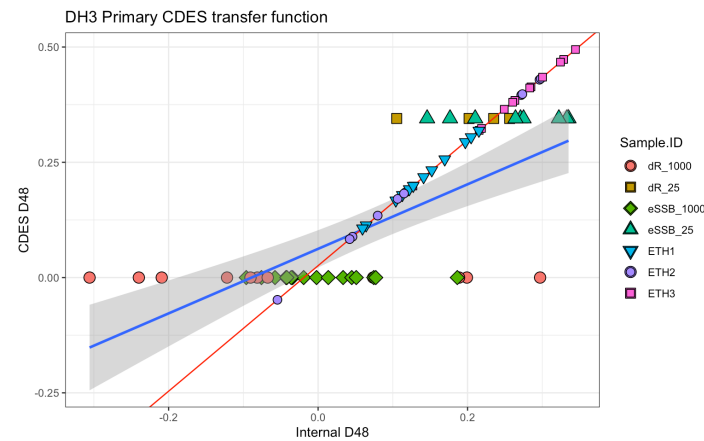
```

```
pCDES48line.pri
```

```

## 'geom_smooth()' using formula 'y ~ x'

```



3. Make heated gas D47 line from data with no D48 excess and no analytical problems

Perform initial regression

```

HG47data <- data %>%
  filter(Donotuse != TRUE & Type=="heated gas" & D48excess.label != "D48.excess") %>%
  select(Sample.ID, Type, num.acq, spec.num, d47, D47, D47.sterr, d47.sterr)

D47line.initial <- cbind(data.frame(Session=session, MS=MS.ID, line.ID="initial D47line"), york.regression(HG47data$d47, HG47data$d47.sterr, HG47data$D47, HG47data$D47.sterr))

HG47data<-calc_exterr(HG47data, 47, D47line.initial$york.slope, D47line.initial$york.intercept, D47line.initial$N)

D47line.final <- cbind(data.frame(Session=session, MS=MS.ID, line.ID="final D47line"), york.regression(HG47data$d47, HG47data$d47.exterr, HG47data$D47, HG47data$D47.exterr))

D47lines<-rbind(D47line.initial, D47line.final)
print(D47lines)

```

```

## Session MS line.ID york.slope york.slope.err york.intercept
## 1 DH3 Bert initial D47line 0.006230944 3.515821e-05 -0.8848291
## 2 DH3 Bert final D47line 0.006189425 1.820255e-04 -0.8838309
## york.int.err MSWD N Cor.ab error.corr
## 1 0.001721607 5.7355525 28 2.788953e-08 0.8562616
## 2 0.008460122 0.2321457 28 5.307274e-07 0.8695246

```

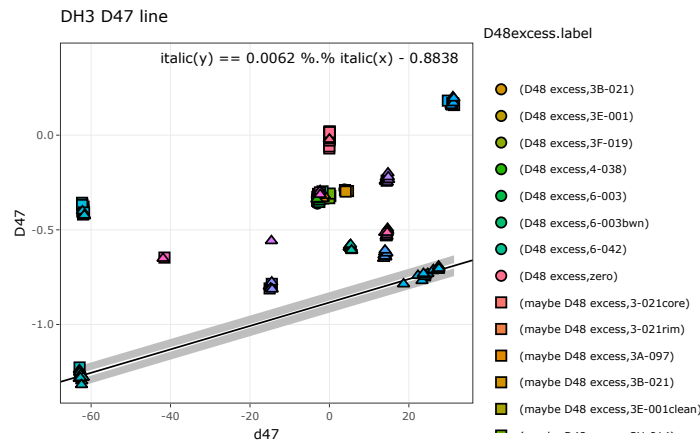
Plot heated gas 47 line with confidence and predictive intervals, including samples w 48 excess

```
D47data <- data %>%
  select(Sample.ID, num.acq, spec.num, Type, Donotuse, D48excess.label, d47, D47, D47.sterr, d47.sterr)

D47data<-york.conf.pred.interval(HG47data, D47data, 47, D47line.final$york.slope, D47line.final$york.intercept,
0.95)

D47line<- D47data %>%
  filter(Donotuse != TRUE) %>%
  ggplot(aes(x=d47, y=D47)) +
  geom_ribbon(aes(ymin=all.predicted.Y-D47.pred.int, ymax=all.predicted.Y+D47.pred.int), fill="gray") +
  geom_ribbon(aes(ymin=all.predicted.Y-D47.conf.int, ymax=all.predicted.Y+D47.conf.int), fill="white") +
  geom_abline(data=D47line.final, aes(slope=york.slope, intercept=york.intercept), size=0.5) +
  geom_point(aes(fill=Sample.ID, shape=D48excess.label), size=3) +
  guides(fill=FALSE) +
  scale_shape_manual(values=c(21,22,24,25,23)) +
  annotate("text", x = -5, y = 0.4, label = york_eqn(D47line.final$york.slope, D47line.final$york.intercept), si
ze = 4, hjust=0, vjust=0, parse=TRUE) +
  theme_bw() +
  labs(title=paste0(session, " D47 line"))

ggplotly(D47line)
```



4. Calculate HG line external D47 errors (includes error from heated gas line)

Calculating final D47 external variance and standard error based on heated gas line

```
data <- calc_prop_errors_lines(data, data$d47, data$d47.sterr, data$d47.sterr, D47line.final$york.slope, D47line.
final$york.slope.err, D47line.final$york.int.err, D47line.final$Cor.ab)
data <- rename(data, D47.HG.prop.err = Y.prop.err) #output from function is meant to be generic: "Y.prop.err" so
this renames the output column to be specific
```

5. Initial data corrections: HG line, stretching, internal acid digestion correction

heated gas, stretching and acid digestion corrections to use for internal Caltech corrections and the gas approach to ARF conversion

```
data$HGcorr <- data$d47 - (data$d47 * D47line.final$york.slope + D47line.final$york.intercept) # HG line correcti
on; this column should be used for conversion to CDES reference frame if using gases to convert
```

6. Absolute Reference Frame (CDES)

6.1. Make ETF lines from data with no D48 excess and calculate CDES D47 values

6.1.1 Primary Approach - uses heated and equilibrated gases

```
CDESline.pri <- data %>%
  filter(D48excess.label=="no D48 excess" & Donotuse != TRUE & Type=="heated gas" | Type=="equilibrated gas") %>%
  left_join(select(Standards, Sample.ID, D47.CDES.90.acc), by = "Sample.ID")
```

```
## Warning: Column `Sample.ID` joining factors with different levels, coercing to
## character vector
```

```
CDES.slope.pri <- (coef(lm(CDESline.pri$d47.CDES.90.acc ~ CDESline.pri$HGcorr)))[2])
CDES.intercept.pri <- (coef(lm(CDESline.pri$d47.CDES.90.acc ~ CDESline.pri$HGcorr)))[1])
```

```
data$CDES.noAD.pri <- CDES.slope.pri*data$HGcorr+CDES.intercept.pri #calculates pre-Acid digestion corrected CDES
values
```

```
#Adds acid digestion factor to CDES values - multiple options: 0.092 is from Henkes et al., 2013. 0.082 is from D
efliese et al., 2015, 0.088 is from Peterson et al., 2019
data$CDES.ADH.pri <- data$CDES.noAD.pri + 0.092
data$CDES.ADD.pri <- data$CDES.noAD.pri + 0.082
data$CDES.ADP.pri <- data$CDES.noAD.pri + 0.088
```

Plots ARF transfer function line from just the heated gas data - the regression in this plot is based only on the heated and equilibrated gas data. The larger standard symbols show the internal heated gas corrected values versus the ARF values they have using the results of this regression, and the smaller sample values show the internal Acid digestion corrected values versus the ARF acid digestion corrected values.

CHECK: if there are no problems with the acid digestion step, AD-corrected values should fall on the line; IS THIS REALLY TRUE?

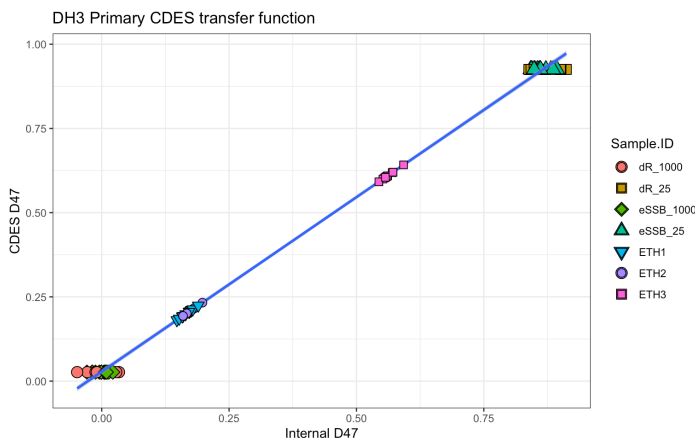
*Note that in this plot, the line is regressed based on ONLY the gas data. The big standards are the pre-acid digestion corrected values, and the small standards are the acid digestion corrected values....

```
pCDESline.pri <- ggplot(CDESline.pri, aes(x=HGcorr, y=D47.CDES.90.acc)) +
  geom_point(aes(fill=Sample.ID, shape=Sample.ID), size=4) +
  stat_smooth(method="lm") +
  geom_point(data=subset(data, Type=="standard" & D48excess.label=="no D48 excess"), aes(x=HGcorr, y=CDES.noAD.pri, fill=Sample.ID, shape=Sample.ID, show_guide=FALSE), size=3) +
  scale_shape_manual(values=c(21,22,23,24,25,21,22,23,24)) +
  labs(title=paste0(session, " Primary CDES transfer function"), x="Internal D47", y="CDES D47") +
  theme_bw()
```

```
## Warning: Ignoring unknown aesthetics: show_guide
```

```
pCDESLine.pri
```

```
## `geom_smooth()` using formula 'y ~ x'
```



6.2.1 Calculate standard residuals from ARF D47 values

Determines standard residuals for all standards, and determines average residuals from standards with no D48 excess; currently using the DeFleise AFF

```
CDES.std.pri <- data %>%
  filter(D48excess.label=="no D48 excess" & Donotuse != TRUE & Type=="standard") # make a subset of HG and EG with
  no analytical problems and no D48 excess
```

```
CDES.std.pri <- left_join(CDES.std.pri, select(Standards, Sample.ID, D47.CDES.90.acc, D47.CDES.25H.acc, D47.CDE
S.25D.acc, D47.CDES.25P.acc), by = "Sample.ID")
```

```
## Warning: Column `Sample.ID` joining factors with different levels, coercing to
## character vector
```

```
CDES.stres.pri<- CDES.std.pri %>%
  group_by(Sample.ID) %>%
  summarise(Residual = "CDES primary",
            Residual.mean.90 = round(mean(CDES.noAD.pri), 4)-D47.CDES.90.acc[1],
            Residual.mean.D = round(mean(CDES.ADD.pri), 4)-D47.CDES.25D.acc[1],
            Residual.mean.H = round(mean(CDES.ADH.pri), 4)-D47.CDES.25H.acc[1],
            Residual.mean.P = round(mean(CDES.ADP.pri), 4)-D47.CDES.25P.acc[1],
            Residual.stdev = round(sd(CDES.noAD.pri), 4),
            Residual.sterr = round(sd(CDES.noAD.pri)/sqrt(length(Sample.ID)), 4),
            N = length(Sample.ID)
  )

CDES.stres.pri
```

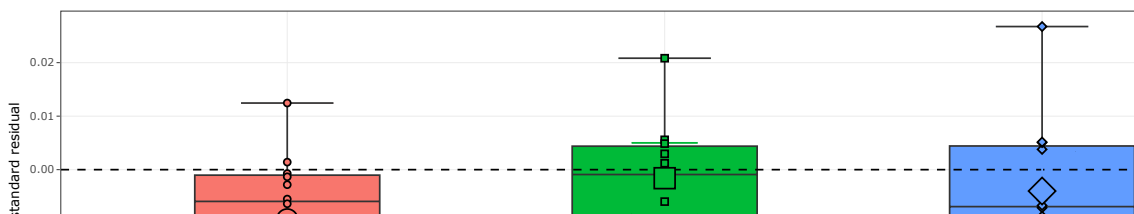
```
## # A tibble: 3 x 9
##   Sample.ID Residual Residual.mean.90 Residual.mean.D Residual.mean.H
##   <chr>      <chr>          <dbl>          <dbl>          <dbl>
## 1 ETH1      CDES pr...    -0.0093      -0.00930      -0.00930
## 2 ETH2      CDES pr...    -0.00160     -0.00160     -0.00160
## 3 ETH3      CDES pr...    -0.004       -0.004       -0.004
## # ... with 4 more variables: Residual.mean.P <dbl>, Residual.stdev <dbl>,
## #   Residual.sterr <dbl>, N <int>
```

Plots of average and 1sigma standard deviation of the standard residuals for standards without D48 excess, with individual values shown, and with and without boxplots superimposed

```
#with boxplots, no acid digestion correction
CDES.residuals.pri.90<-ggplot(CDES.stres.pri, aes(x=Sample.ID, y=Residual.mean.90, shape=Sample.ID, fill=Sample.
ID)) +
  geom_hline(yintercept=0, linetype=2) +
  geom_errorbar(aes(ymin=(Residual.mean.90-2*Residual.sterr), ymax=(Residual.mean.90+2*Residual.sterr), colour=Sa
mple.ID), width=.3) +
  geom_boxplot(data=CDES.std.pri, aes(x=Sample.ID, y=CDES.noAD.pri-D47.CDES.90.acc, fill=Sample.ID, shape=NULL))
+
  geom_point(data=CDES.std.pri, aes(x=Sample.ID, y=CDES.noAD.pri-D47.CDES.90.acc, fill=Sample.ID, shape=Sample.I
D), size=2) +
  geom_point(data=CDES.stres.pri, aes(x=Sample.ID, y=Residual.mean.90, fill=Sample.ID, shape=Sample.ID), size=6)
+
  scale_shape_manual(values=c(21,22,23,24,25)) +
  labs(title=paste0(session, " avg. and 2se, boxplot of residuals, CDES primary"), x="Standard ID", y="Mean stand
ard residual") +
  theme_bw() +
  theme(legend.title=element_blank())

ggplotly(CDES.residuals.pri.90)
```

DH3 avg. and 2se, boxplot of residuals, CDES primary



6.3.1 Standard corrections for CDES D47 values from data with no D48 excess

6.3.1.a Standard correct using a line between the standards

```
#plot line between standards for correcting data for no acid digestion values, Henkes AFF values and Defliese AFF
values

CDES.stds.pri <- CDES.std.pri %>%
  gather(CDES_type, CDES_values, CDES.noAD.pri, CDES.ADH.pri, CDES.ADD.pri, CDES.ADP.pri) %>%
  gather(CDES_acc_type, CDES_acc_values, D47.CDES.90.acc, D47.CDES.25H.acc, D47.CDES.25D.acc, D47.CDES.25P.acc) %
  >%
  filter(CDES_type == "CDES.noAD.pri" & CDES_acc_type == "D47.CDES.90.acc" | CDES_type == "CDES.ADH.pri" & CDES_a
cc_type == "D47.CDES.25H.acc" | CDES_type == "CDES.ADD.pri" & CDES_acc_type == "D47.CDES.25D.acc" | CDES_type ==
"CDES.ADP.pri" & CDES_acc_type == "D47.CDES.25P.acc")

CDES.std.line.pri<-ggplot(CDES.stds.pri) +
  geom_abline(slope=1, intercept=0, linetype=2) +
  stat_smooth(method = "lm", aes(x=CDES_values, y=CDES_acc_values, colour = CDES_type)) +
  geom_point(aes(x=CDES_values, y=CDES_acc_values, fill = CDES_type, shape = Sample.ID), size=4) +
  labs(title=paste0(session, " line for CDES.std correction, primary CDES")) +
  scale_shape_manual(values = c(21,22,23,24,25)) +
  theme_bw()

#standard correct the Henkes AFF values
CDES.std.line.slope.pri.H<-(coef(lm(CDES.std.pri$D47.CDES.25H.acc ~ CDES.std.pri$CDES.ADH.pri)))[2]) #calculates
slope of measured v accepted line btw standards
CDES.std.line.intercept.pri.H<-(coef(lm(CDES.std.pri$D47.CDES.25H.acc ~ CDES.std.pri$CDES.ADH.pri)))[1]) #calcs
intercept of measured v accepted standard line
CDES.std.line.offset.pri.H<-data$CDES.ADH.pri - (CDES.std.line.slope.pri.H * data$CDES.ADH.pri + CDES.std.line.in
tercept.pri.H) #calculates offset values for all acid corrected values

data$CDES.D47stdcorrH.pri<-data$CDES.ADH.pri - CDES.std.line.offset.pri.H #calculates D47 values using the standa
rd line correction

#standard correct the Defliese AFF values
CDES.std.line.slope.pri.D<-(coef(lm(CDES.std.pri$D47.CDES.25D.acc ~ CDES.std.pri$CDES.ADD.pri)))[2]) #calculates
slope of measured v accepted line btw standards
CDES.std.line.intercept.pri.D<-(coef(lm(CDES.std.pri$D47.CDES.25D.acc ~ CDES.std.pri$CDES.ADD.pri)))[1]) #calcs
intercept of measured v accepted standard line
CDES.std.line.offset.pri.D<-data$CDES.ADD.pri - (CDES.std.line.slope.pri.D * data$CDES.ADD.pri + CDES.std.line.in
tercept.pri.D) #calculates offset values for all acid corrected values

data$CDES.D47stdcorrD.pri<-data$CDES.ADD.pri - CDES.std.line.offset.pri.D #calculates D47 values using the standa
rd line correction

#standard correct the Peterson AFF values
CDES.std.line.slope.pri.P<-(coef(lm(CDES.std.pri$D47.CDES.25P.acc ~ CDES.std.pri$CDES.ADP.pri)))[2]) #calculates
slope of measured v accepted line btw standards
CDES.std.line.intercept.pri.P<-(coef(lm(CDES.std.pri$D47.CDES.25P.acc ~ CDES.std.pri$CDES.ADP.pri)))[1]) #calcs
intercept of measured v accepted standard line
CDES.std.line.offset.pri.P<-data$CDES.ADP.pri - (CDES.std.line.slope.pri.P * data$CDES.ADP.pri + CDES.std.line.in
tercept.pri.P) #calculates offset values for all acid corrected values

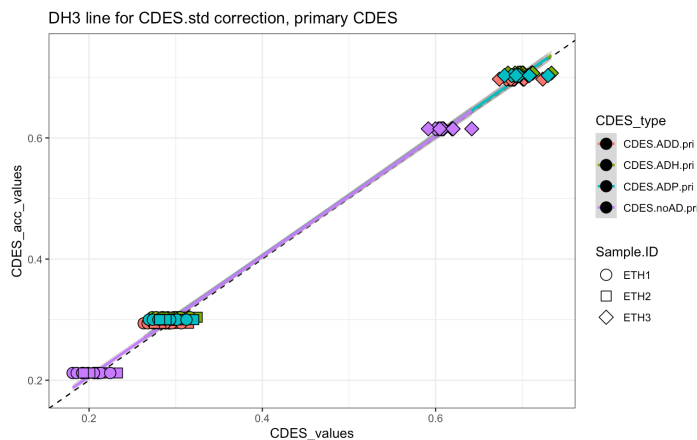
data$CDES.D47stdcorrP.pri<-data$CDES.ADP.pri - CDES.std.line.offset.pri.P #calculates D47 values using the standa
rd line correction

#standard correct the no-acid-correction values (for Bonifacie calibration)
CDES.std.line.slope.pri.noAD<-(coef(lm(CDES.std.pri$D47.CDES.90.acc ~ CDES.std.pri$CDES.noAD.pri)))[2]) #calcula
tes slope of measured v accepted line btw standards
CDES.std.line.intercept.pri.noAD<-(coef(lm(CDES.std.pri$D47.CDES.90.acc ~ CDES.std.pri$CDES.noAD.pri)))[1]) #cal
cs intercept of measured v accepted standard line
CDES.std.line.offset.pri.noAD<-data$CDES.noAD.pri - (CDES.std.line.slope.pri.noAD * data$CDES.noAD.pri + CDES.st
d.line.intercept.pri.noAD) #calculates offset values for all acid corrected values

data$CDES.D47stdcorr90.pri <-data$CDES.noAD.pri - CDES.std.line.offset.pri.noAD #calculates D47 values using the
standard line correction

CDES.std.line.pri
```

```
## `geom_smooth()` using formula 'y ~ x'
```



6.4.1 Calculate temperatures

```
data$CDES.ghosh.temp.pri.H <- convert_ARF.D47_to_ARF.Ghosh.temp(data$CDES.ADH.pri) #calculate temps from Acid corrected D47 values (Henkes AFF)
data$CDES.ghosh.temp.stdcorr.pri.H <- convert_ARF.D47_to_ARF.Ghosh.temp(data$CDES.D47stdcorrH.pri) #calculate temps from Acid corrected (Henkes AFF) and carbonate standard corrected D47 values

data$CDES.ghosh.temp.pri.D <- convert_ARF.D47_to_ARF.Ghosh.temp(data$CDES.ADD.pri) #calculate temps from Acid corrected D47 values (Defliese AFF)
data$CDES.ghosh.temp.stdcorr.pri.D <- convert_ARF.D47_to_ARF.Ghosh.temp(data$CDES.D47stdcorrD.pri) #calculate temps from Acid corrected (Defliese AFF) and carbonate standard corrected D47 values

data$CDES.peterson.temp.pri <- convert_ARF.D47_to_ARF.Peterson.temp(data$CDES.ADP.pri) #calculate temps from Acid corrected D47 values (Peterson AFF)
data$CDES.peterson.temp.stdcorr.pri <- convert_ARF.D47_to_ARF.Peterson.temp(data$CDES.D47stdcorrP.pri) #calculate temps from Acid corrected (Peterson AFF) and carbonate standard corrected D47 values

data$CDES.Bonifacie.temp.pri <- convert_ARF.D47_to_ARF.Bonifacie.temp(data$CDES.noAD.pri) #calculate temps from Acid corrected D47 values (no AFF)
data$CDES.Bonifacie.temp.stdcorr.pri <- convert_ARF.D47_to_ARF.Bonifacie.temp(data$CDES.D47stdcorr90.pri) #calculate temps from no Acid corrected values and carbonate standard corrected D47 values
```

6.1. Make ETF lines from data with no D48 excess and calculate ARF D47 values

6.1.2 Secondary Approach -uses standards to make ETF

```
CDESLine.sec <- data %>%
  filter(D48excess.label=="no D48 excess" & Donotuse != TRUE & Type=="standard") %>%
  left_join(select(Standards, Sample.ID, D47.CDES.90.acc), by = "Sample.ID")
```

```
## Warning: Column 'Sample.ID' joining factors with different levels, coercing to
## character vector
```

```
CDES.slope.sec <- (coef(lm(CDESLine.sec$D47.CDES.90.acc ~ CDESLine.sec$HGcorr)))[2])
CDES.intercept.sec <- (coef(lm(CDESLine.sec$D47.CDES.90.acc ~ CDESLine.sec$HGcorr)))[1])

data$CDES.noAD.sec <- CDES.slope.sec+data$HGcorr+CDES.intercept.sec #calculates pre-Acid digestion corrected CDES values

data$CDES.ADH.sec <- data$CDES.noAD.sec + 0.092 #Adds acid digestion factor to CDES values - multiple options: 0.092 is from Henkes et al., 2013. 0.082 is from Defliese et al., 2015.

data$CDES.ADD.sec <- data$CDES.noAD.sec + 0.082 #Adds acid digestion factor to CDES values - multiple options: 0.092 is from Henkes et al., 2013. 0.082 is from Defliese et al., 2015.
```

Plots ARF transfer function line from just the heated gas data - the regression in this plot is based only on the heated and equilibrated gas data. The larger standard symbols show the internal heated gas corrected values versus the ARF values they have using the results of this regression, and the smaller sample values show the internal Acid digestion corrected values versus the ARF acid digestion corrected values.

CHECK: if there are no problems with the acid digestion step, AD-corrected values should fall on the line; IS THIS REALLY TRUE?

*Note that in this plot, the line is regressed based on ONLY the gas data. The big standards are the pre-acid digestion corrected values, and the small standards are the acid digestion corrected values....

```
pCDESLineH.sec <- ggplot(CDESLine.sec, aes(x=HGcorr, y=D47.CDES.90.acc)) +
  geom_point(aes(fill=Sample.ID, shape=Sample.ID), size=4) +
  stat_smooth(method="lm") +
  geom_point(data=subset(data, Type=="standard" & D48excess.label=="no D48 excess"), aes(x=HGcorr, y=CDES.noAD.sec, fill=Sample.ID, shape=Sample.ID, show_guide=FALSE), size=3) +
  scale_shape_manual(values=c(21,22,23,24,25,21,22,23,24)) +
  labs(title=paste0(session, " Secondary CDES transfer function, using Henkes AFF"), x="Internal D47", y="CDES D47") +
  theme_bw()
```

```
## Warning: Ignoring unknown aesthetics: show_guide
```

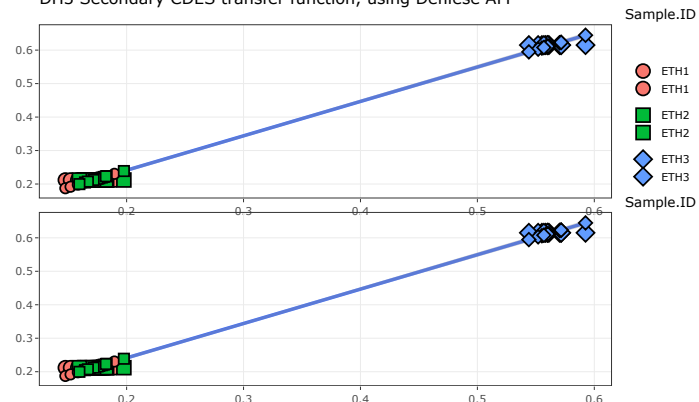
```
pCDESLineD.sec <- ggplot(CDESLine.sec, aes(x=HGcorr, y=D47.CDES.90.acc)) +
  geom_point(aes(fill=Sample.ID, shape=Sample.ID), size=4) +
  stat_smooth(method="lm") +
  geom_point(data=subset(data, Type=="standard" & D48excess.label=="no D48 excess"), aes(x=HGcorr, y=CDES.noAD.sec, fill=Sample.ID, shape=Sample.ID, show_guide=FALSE), size=3) +
  scale_shape_manual(values=c(21,22,23,24,25,21,22,23,24)) +
  labs(title=paste0(session, " Secondary CDES transfer function, using Defliese AFF"), x="Internal D47", y="ARF D47") +
  theme_bw()
```

```
## Warning: Ignoring unknown aesthetics: show_guide
```

```
subplot(pCDESLineH.sec, pCDESLineD.sec, nrows = 2)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

DH3 Secondary CDES transfer function, using Defliese AFF



6.2.2 Calculate standard residuals from ARF D47 values

Determines standard residuals for all standards, and determines average residuals from standards with no D48 excess; currently using the DeFleise AFF

```
CDES.std.sec <- data %>%
  filter(D48excess.label=="no D48 excess" & Donotuse != TRUE & Type=="standard") # make a subset of HG and EG wit
  # no analytical problems and no D48 excess

CDES.std.sec <- left_join(CDES.std.sec, select(Standards, Sample.ID, D47.CDES.90.acc, D47.CDES.25H.acc, D47.CDE
S.25D.acc), by = "Sample.ID")
```

```
## Warning: Column 'Sample.ID' joining factors with different levels, coercing to
## character vector
```

```
CDES.stdres.sec<- CDES.std.sec %>%
  group_by(Sample.ID) %>%
  summarise(Residual = "CDES secondary",
            Residual.mean.90 = round(mean(CDES.noAD.sec), 4)-D47.CDES.90.acc[1],
            Residual.mean.D=round(mean(CDES.ADD.sec), 4)-D47.CDES.25D.acc[1],
            Residual.mean.H=round(mean(CDES.ADH.sec), 4)-D47.CDES.25H.acc[1],
            Residual.stdev=round(sd(CDES.ADH.sec), 4),
            Residual.sterr=round(sd(CDES.ADH.sec)/sqrt(length(Sample.ID)), 4),
            N= length(Sample.ID)
  )

CDES.stdres.sec
```

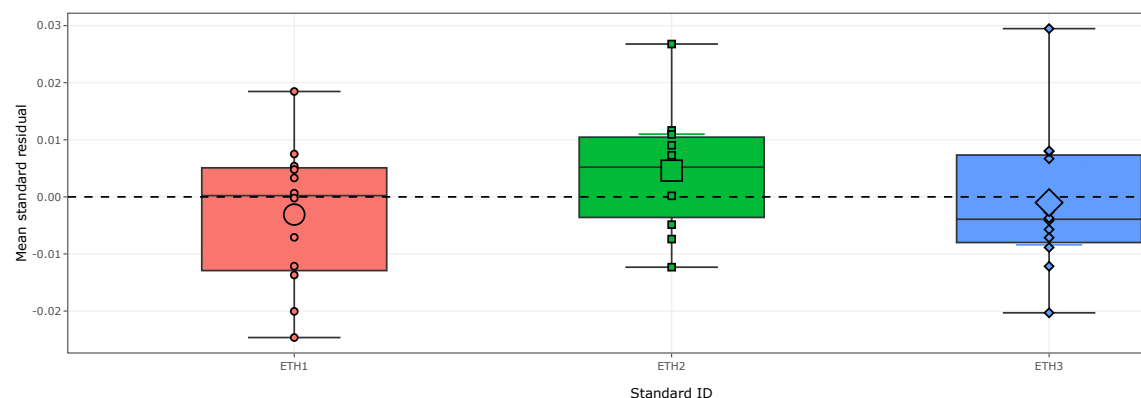
```
## # A tibble: 3 x 8
##   Sample.ID Residual Residual.mean.90 Residual.mean.D Residual.mean.H
##   <chr>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 ETH1      CDES se...    -0.00310    -0.00310    -0.00310
## 2 ETH2      CDES se...     0.00460     0.00460     0.00460
## 3 ETH3      CDES se...    -0.001     -0.001     -0.001
## # ... with 3 more variables: Residual.stdev <dbl>, Residual.sterr <dbl>, N <int>
```

Plots of average and 1sigma standard deviation of the standard residuals for standards without D48 excess, with individual values shown, and with and without boxplots superimposed

```
#with boxplots, no acid digestion correction
CDES.residuals.sec.bp.90<-ggplot(CDES.stdres.sec, aes(x=Sample.ID, y=Residual.mean.90, shape=Sample.ID, fill=Sam
ple.ID)) +
  geom_hline(yintercept=0, linetype=2) +
  geom_errorbar(aes(ymin=(Residual.mean.90-2*Residual.sterr), ymax=(Residual.mean.90+2*Residual.sterr), colour=Sa
mple.ID), width=.3) +
  geom_boxplot(data=CDES.std.sec, aes(x=Sample.ID, y=CDES.noAD.sec-D47.CDES.90.acc, fill=Sample.ID, shape=NULL))
+
  geom_point(data=CDES.std.sec, aes(x=Sample.ID, y=CDES.noAD.sec-D47.CDES.90.acc, fill=Sample.ID, shape=Sample.I
D), size=2) +
  geom_point(data=CDES.stdres.sec, aes(x=Sample.ID, y=Residual.mean.90, fill=Sample.ID, shape=Sample.ID), size=6)
+
  scale_shape_manual(values=c(21,22,23,24,25)) +
  labs(title=paste0(session, " avg. and 2se, boxplot of residuals, CDES secondary"), x="Standard ID", y="Mean sta
ndard residual") +
  theme_bw() +
  theme(legend.title=element_blank())

ggplotly(CDES.residuals.sec.bp.90)
```

DH3 avg. and 2se, boxplot of residuals, CDES secondary



6.3.2 Standard corrections for CDES D47 values from data with no D48 excess

6.3.2.a Standard correct using a line between the standards

```

#plot line between standards for correcting data for no acid digestion values, Henkes AFF values and Defliese AFF values

CDES.stds.sec <- CDES.std.sec %>%
  gather(CDES_type, CDES_values, CDES.noAD.sec, CDES.ADH.sec, CDES.ADD.sec) %>%
  gather(CDES_acc_type, CDES_acc_values, D47.CDES.90.acc, D47.CDES.25H.acc, D47.CDES.25D.acc) %>%
  filter(CDES_type == "CDES.noAD.sec" & CDES_acc_type == "D47.CDES.90.acc" | CDES_type == "CDES.ADH.sec" & CDES_acc_type == "D47.CDES.25H.acc" | CDES_type == "CDES.ADD.sec" & CDES_acc_type == "D47.CDES.25D.acc")

CDES.std.line.sec<-ggplot(CDES.stds.sec) +
  geom_abline(slope=1, intercept=0, linetype=2) +
  stat_smooth(method = "lm", aes(x=CDES_values, y=CDES_acc_values, colour = CDES_type)) +
  geom_point(aes(x=CDES_values, y=CDES_acc_values, fill = CDES_type, shape = Sample.ID), size=4) +
  labs(title=paste0(session, " line for CDES.std correction, secondary CDES")) +
  scale_shape_manual(values = c(21,22,23,24,25)) +
  theme_bw()

#standard correct the Henkes AFF values
CDES.std.line.slope.sec.H<-(coef(lm(CDES.std.sec$D47.CDES.25H.acc ~ CDES.std.sec$CDES.ADH.sec)))[2]) #calculates slope of measured v accepted line btw standards
CDES.std.line.intercept.sec.H<-(coef(lm(CDES.std.sec$D47.CDES.25H.acc ~ CDES.std.sec$CDES.ADH.sec)))[1]) #calcs intercept of measured v accepted standard line
CDES.std.line.offset.sec.H<-data$CDES.ADH.sec - (CDES.std.line.slope.sec.H * data$CDES.ADH.sec + CDES.std.line.intercept.sec.H) #calculates offset values for all acid corrected values

data$CDES.D47stdcorrH.sec<-data$CDES.ADH.sec - CDES.std.line.offset.sec.H #calculates D47 values using the standard line correction

#standard correct the Defliese AFF values
CDES.std.line.slope.sec.D<-(coef(lm(CDES.std.sec$D47.CDES.25D.acc ~ CDES.std.sec$CDES.ADD.sec)))[2]) #calculates slope of measured v accepted line btw standards
CDES.std.line.intercept.sec.D<-(coef(lm(CDES.std.sec$D47.CDES.25D.acc ~ CDES.std.sec$CDES.ADD.sec)))[1]) #calcs intercept of measured v accepted standard line
CDES.std.line.offset.sec.D<-data$CDES.ADD.sec - (CDES.std.line.slope.sec.D * data$CDES.ADD.sec + CDES.std.line.intercept.sec.D) #calculates offset values for all acid corrected values

data$CDES.D47stdcorrD.sec<-data$CDES.ADD.sec - CDES.std.line.offset.sec.D #calculates D47 values using the standard line correction

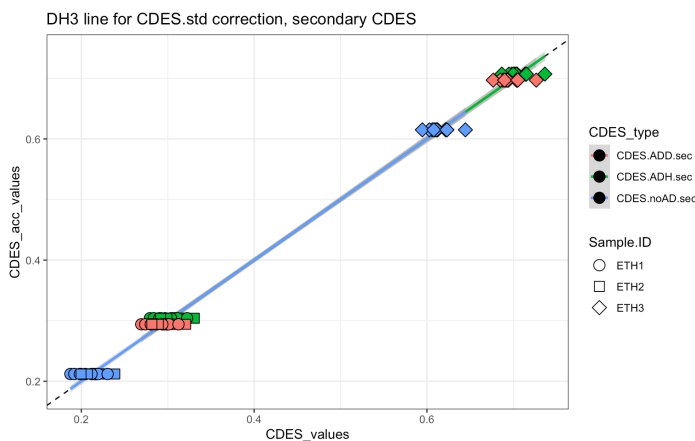
#standard correct the no-acid-correction values (for Bonifacie calibration)
CDES.std.line.slope.sec.noAD<-(coef(lm(CDES.std.sec$D47.CDES.90.acc ~ CDES.std.sec$CDES.noAD.sec)))[2]) #calculates slope of measured v accepted line btw standards
CDES.std.line.intercept.sec.noAD<-(coef(lm(CDES.std.sec$D47.CDES.90.acc ~ CDES.std.sec$CDES.noAD.sec)))[1]) #calcs intercept of measured v accepted standard line
CDES.std.line.offset.sec.noAD<-data$CDES.noAD.sec - (CDES.std.line.slope.sec.noAD * data$CDES.noAD.sec + CDES.std.line.intercept.sec.noAD) #calculates offset values for all acid corrected values

data$CDES.D47stdcorr90.sec<-data$CDES.noAD.sec - CDES.std.line.offset.sec.noAD #calculates D47 values using the standard line correction

CDES.std.line.sec

```

```
## 'geom_smooth()' using formula 'y ~ x'
```



6.4.2 Calculate temperatures

```

data$CDES.ghosh.temp.sec.H <- convert_ARF.D47_to_ARF.Ghosh.temp(data$CDES.ADH.sec) #calculate temps from Acid corrected D47 values (Henkes AFF)
data$CDES.ghosh.temp.stdcorr.sec.H <- convert_ARF.D47_to_ARF.Ghosh.temp(data$CDES.D47stdcorrH.sec) #calculate temps from Acid corrected (Henkes AFF) and carbonate standard corrected D47 values

data$CDES.ghosh.temp.sec.D <- convert_ARF.D47_to_ARF.Ghosh.temp(data$CDES.ADD.sec) #calculate temps from Acid corrected D47 values (Defliese AFF)
data$CDES.ghosh.temp.stdcorr.sec.D <- convert_ARF.D47_to_ARF.Ghosh.temp(data$CDES.D47stdcorrD.sec) #calculate temps from Acid corrected (Defliese AFF) and carbonate standard corrected D47 values

data$CDES.Bonifacie.temp.sec <- convert_ARF.D47_to_ARF.Bonifacie.temp(data$CDES.noAD.sec) #calculate temps from Acid corrected D47 values (no AFF)
data$CDES.Bonifacie.temp.stdcorr.sec <- convert_ARF.D47_to_ARF.Bonifacie.temp(data$CDES.D47stdcorr90.sec) #calculate temps from no Acid corrected values and carbonate standard corrected D47 values

```

Data processing synthesis and comparisons

Synthesis tables

D48 line parameters for 1st and 2nd methods; will overwrite previous versions of the sample session data if an earlier round was already appended

```

York.lines<-rbind(D48lines, D47lines)
print(York.lines)

```

```
## Session MS line.ID york.slope york.slope.err york.intercept
## 1 DH3 Bert initial D48line 0.071884586 2.386609e-04 -0.3819867
## 2 DH3 Bert final D48line 0.071809200 5.929465e-04 -0.3757578
## 3 DH3 Bert initial D47line 0.006230944 3.515821e-05 -0.8848291
## 4 DH3 Bert final D47line 0.006189425 1.820255e-04 -0.8838309
## york.int.err MSWD N Cor.ab error.corr
## 1 0.007387157 20.5208347 28 9.174859e-07 0.9845513
## 2 0.018463839 3.4477905 28 5.772863e-06 0.9856182
## 3 0.001721607 5.7355525 28 2.788953e-08 0.8562616
## 4 0.008460122 0.2321457 28 5.307274e-07 0.8695246
```

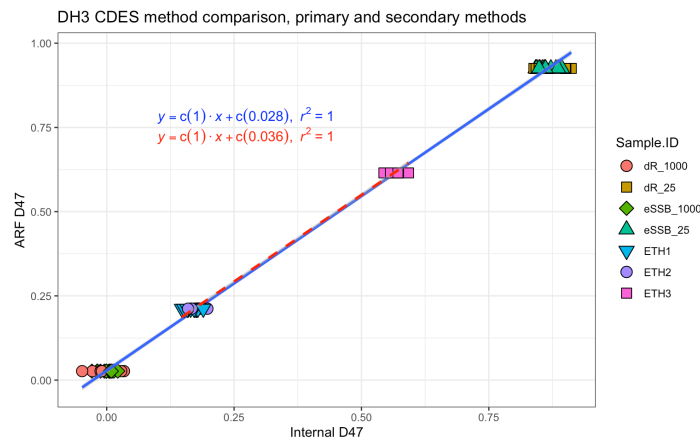
```
write.csv(York.lines, file=paste0(session, "_Yorklineparameters.csv"))
```

Comparison of primary and secondary

```
CDESLines<-ggplot(CDESLine.pri, aes(x=HGcorr, y=D47.CDES.90.acc)) +
  geom_point(aes(fill=Sample.ID, shape=Sample.ID), size=4) +
  stat_smooth(method="lm") +
  geom_point(data=CDESLine.sec, aes(x=HGcorr, y=D47.CDES.90.acc, fill=Sample.ID, shape=Sample.ID), size=4) +
  stat_smooth(data=CDESLine.sec, aes(x=HGcorr, y=D47.CDES.90.acc), method="lm", linetype=2, colour="red") +
  scale_shape_manual(values=c(21,22,23,24,25,21,22,23,24)) +
  annotate("text", x = 0.1, y = 0.76, label = lm_eqn(CDESLine.pri$HGcorr, CDESLine.pri$D47.CDES.90.acc), size =
4, hjust=0, vjust=0, parse=TRUE, colour="blue") +
  annotate("text", x = 0.1, y = 0.70, label = lm_eqn(CDESLine.sec$HGcorr, CDESLine.sec$D47.CDES.90.acc), size =
4, hjust=0, vjust=0, parse=TRUE, colour="red") +
  labs(title=paste0(session, " CDES method comparison, primary and secondary methods"), x="Internal D47", y="ARF
D47") +
  theme_bw()
```

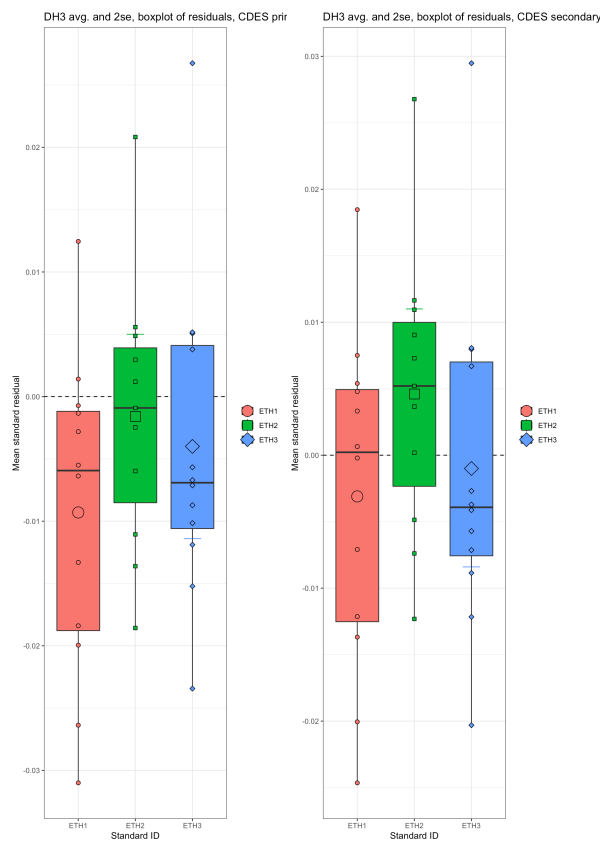
CDESLines

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



Comparison of standard residuals

```
residual.comparison <- multiplot(CDES.residuals.pri.90, CDES.residuals.sec.bp.90, cols=3)
```



```
Std.res.all <- bind_rows(CDES.stdres.pri, CDES.stdres.sec)

write.csv(Std.res.all, file=paste0(session, "_CITstdresiduals.csv"))

residual.comparison
```

```
## NULL
```

standard correcting bulk isotopes values

```
bulk.stds.acc <- read.csv("bulkisostds.csv")

bulk.stds <- data %>%
  filter(D48excess.label!="D48 excess" & Donotuse != TRUE & Type=="standard" | Sample.ID == "NBS18") # make a sub
  set of HG and EG with no analytical problems and no D48 excess

bulk.stds <- left_join(bulk.stds, select(bulk.stds.acc, Sample.ID, d180.acc, d13C.acc), by = "Sample.ID")
```

```
## Warning: Column `Sample.ID` joining factors with different levels, coercing to
## character vector
```

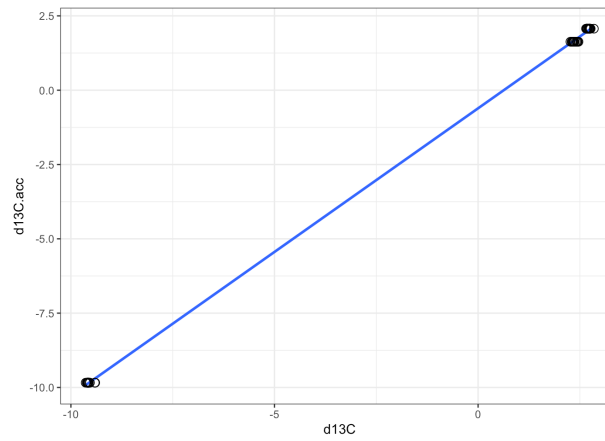
d13C discrimination correction - First, need to pick which correction scheme you prefer, then update this with the correct data columns

```
# regression
m.d13C <- lm(bulk.stds$d13C.acc ~ bulk.stds$d13C)
disc.slopeC<-(coef(m.d13C)[2])
disc.interC<-(coef(m.d13C)[1])
R2 <- summary(m.d13C)$r.squared

disc.plot.C <- ggplot(bulk.stds, aes(x=d13C, y = d13C.acc)) +
  stat_smooth(method = "lm") +
  geom_point(size = 3, shape = 21) +
  theme_bw()

disc.plot.C
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
# apply correction to data
data$d13C.disc.offset <- data$d13C - (disc.slopeC * data$d13C + disc.interC) #calculates offset values for all ac
id corrected values
data$d13C.disc <- data$d13C - data$d13C.disc.offset
```

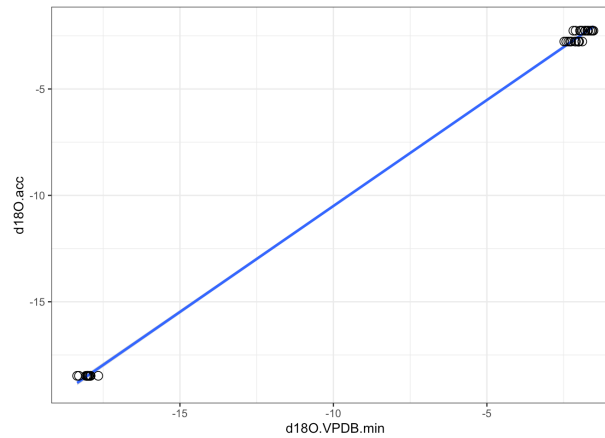
18O discrimination correction - First, need to pick which correction scheme you prefer, then update this with the correct data columns

```
# regression
m.d180 <- lm(bulk.stds$d180.acc ~ bulk.stds$d180.VPDB.min)
disc.slope0<-(coef(m.d180)[[2]])
disc.inter0<-(coef(m.d180)[[1]])
R2 <- summary(m.d180)$r.squared

disc.plot.0 <- ggplot(bulk.stds, aes(x=d180.VPDB.min, y = d180.acc)) +
  stat_smooth(method = "lm") +
  geom_point(size = 3, shape = 21) +
  theme_bw()

disc.plot.0
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
# apply correction to data
data$d180.disc.offset <- data$d180.VPDB.min - (disc.slope0 * data$d180.VPDB.min + disc.inter0) #calculates offset
values for all acid corrected values
data$d180.disc <- data$d180.VPDB.min - data$d180.disc.offset
```

average reference data (standards, equilibrated gases), and plot D47.90 vs D48, 90 and compare to theoretical dual equilibrium line of Fiebig et al., 2019 (derived from Hill et al., 2014 Keq calculations for D63 and D64 of calcite)

```

data.avg <- data %>%
  group_by(Sample.ID) %>%
  filter(D48excess.label != "D48 excess") %>%
  summarise(Project = Project[1],
    N = n(),
    d13C.sd = sqrt(calc_wgt_variance(d13C.disc, d13C.sterr)),
    d13C.se = d13C.sd/sqrt(N),
    d13C.avg = calc_wgt_means(d13C.disc, d13C.sterr),
    d180.sd = sqrt(calc_wgt_variance(d180.disc, d180.sterr)),
    d180.se = d180.sd/sqrt(N),
    d180.avg = calc_wgt_means(d180.disc, d180.sterr),
    D47.sd = sqrt(calc_wgt_variance(CDES.noAD.pri, D47.HG.prop.err)),
    D47.se = D47.sd/sqrt(N),
    D47.2se = 2*D47.se,
    D47.90.noAD.avg = calc_wgt_means(CDES.noAD.pri, D47.HG.prop.err),
    D47.90.avg = calc_wgt_means(CDES.D47stdcorr90.pri, D47.HG.prop.err),
    D47.250.avg = calc_wgt_means(CDES.D47stdcorr0.pri, D47.HG.prop.err),
    D47.25P.avg = calc_wgt_means(CDES.D47stdcorrP.pri, D47.HG.prop.err),
    D48.sd = sqrt(calc_wgt_variance(CDES.48noAD.pri, D48.sterr)),
    D48.se = D48.sd/sqrt(N),
    D48.2se = 2*D48.se,
    D48.90.avg = calc_wgt_means(CDES.48noAD.pri, D48.sterr),
    D48.sd.all = sqrt(calc_wgt_variance(CDES.48noAD.pri.all, D48.sterr)),
    D48.se.all = D48.sd.all/sqrt(N),
    D48.2se.all = 2*D48.se.all,
    D48.90.avg.all = calc_wgt_means(CDES.48noAD.pri.all, D48.sterr),
    T.B = convert_ARF.D47_to_ARF.Bonifacie.temp(D47.90.avg),
    T.B.se = calc_D47.Temp_se(T.B, D47.90.avg, D47.se),
    T.G = convert_ARF.D47_to_ARF.Ghosh.temp(D47.250.avg),
    T.G.se = calc_D47.Temp_se(T.B, D47.250.avg, D47.se),
    T.P = convert_ARF.D47_to_ARF.Peterson.temp(D47.25P.avg),
    T.P.se = calc_D47.Temp_se(T.P, D47.25P.avg, D47.se),
    d180w.B = calc_d180w(d180.avg, T.B),
    d180w.B.se = calc_d180wse(d180.avg, d180.se, T.B, T.B.se),
    d180w.G = calc_d180w(d180.avg, T.G),
    d180w.G.se = calc_d180wse(d180.avg, d180.se, T.G, T.G.se),
    d180w.P = calc_d180w(d180.avg, T.P),
    d180w.P.se = calc_d180wse(d180.avg, d180.se, T.P, T.P.se))

```

```
## Warning in sqrt(calc_wgt_variance(d13C.disc, d13C.sterr)): NaNs produced
```

```
## Warning in sqrt(calc_wgt_variance(d180.disc, d180.sterr)): NaNs produced
```

```
## Warning in sqrt(calc_wgt_variance(CDES.noAD.pri, D47.HG.prop.err)): NaNs
## produced
```

```

pD47vD48 <- ggplot(data = data.frame(x=c(0.1, 0.7)), aes(x)) +
  stat_function(fun = function(x) {0.1829*(x^2) + 0.2138*(x) - 0.0139}, colour = "light gray") +
  stat_function(fun = function(x) {0.3821*(x^2) + 0.0044*(x) + 0.1198}) +
  geom_errorbar(data = data.avg, aes(x=D47.90.noAD.avg, ymin=D48.90.avg - D48.2se, ymax=D48.90.avg + D48.2se, col
our = Sample.ID)) +
  geom_errorbarh(data = data.avg, aes(x=D47.90.noAD.avg, y=D48.90.avg, xmin=D47.90.noAD.avg - D47.2se, xmax = D4
7.90.noAD.avg + D47.2se , colour = Sample.ID)) +
  geom_point(data = data.avg, aes(x=D47.90.noAD.avg, y=D48.90.avg, fill = Sample.ID, colour = NULL), size = 3, sh
ape = 21) +
  #scale_shape_manual(values = c(21,22,23,24,25)) +
  theme_bw()

```

```
## Warning: Ignoring unknown aesthetics: x
```

```

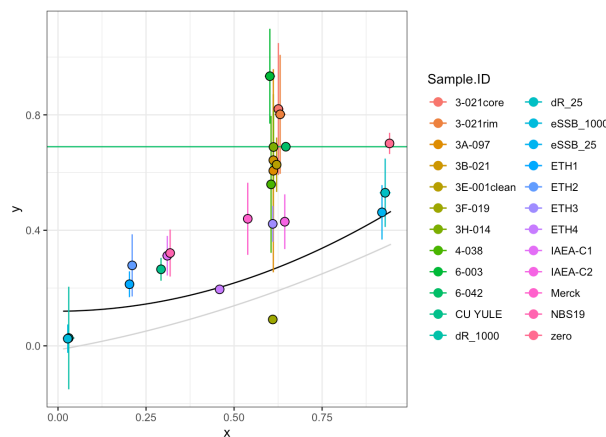
pD47vD48.all <- ggplot(data = data.frame(x=c(0.1, 0.7)), aes(x)) +
  stat_function(fun = function(x) {0.1829*(x^2) + 0.2138*(x) - 0.0139}, colour = "light gray") +
  stat_function(fun = function(x) {0.3821*(x^2) + 0.0044*(x) + 0.1198}) +
  geom_errorbar(data = data.avg, aes(x=D47.90.noAD.avg, ymin=D48.90.avg.all - D48.2se.all, ymax=D48.90.avg.all +
D48.2se.all, colour = Sample.ID)) +
  geom_errorbarh(data = data.avg, aes(x=D47.90.noAD.avg, y=D48.90.avg.all, xmin=D47.90.noAD.avg - D47.2se, xmax =
D47.90.noAD.avg + D47.2se , colour = Sample.ID)) +
  geom_point(data = data.avg, aes(x=D47.90.noAD.avg, y=D48.90.avg.all, fill = Sample.ID, colour = NULL), size =
3, shape = 21) +
  #scale_shape_manual(values = c(21,22,23,24,25)) +
  theme_bw()

```

```
## Warning: Ignoring unknown aesthetics: x
```

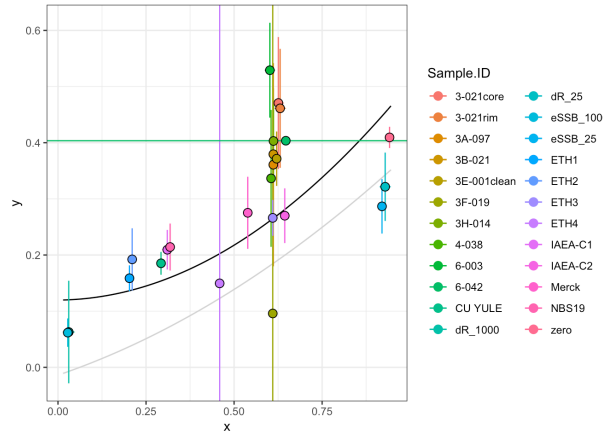
```
pD47vD48
```

```
## Warning: Removed 2 rows containing missing values (geom_errorbarh).
```



pD47vD48.all

Warning: Removed 2 rows containing missing values (geom_errorbarh).



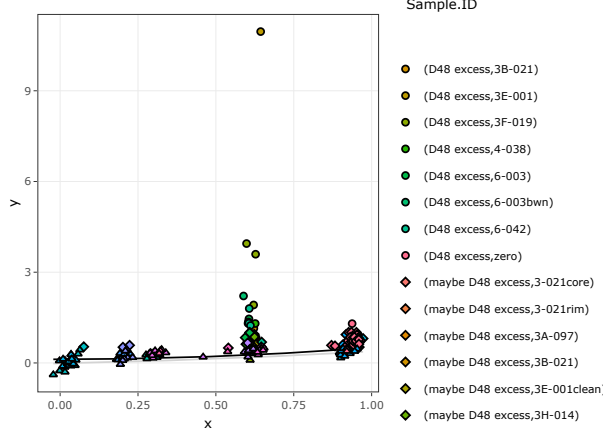
data.avg

```
## # A tibble: 24 x 36
##   Sample.ID Project N d13C.sd d13C.se d13C.avg d180.sd d180.se
##   <fct> <fct> <int> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 3-021core WP2011 4 0.0757 0.0378 -5.96 0.148 0.0738
## 2 3-021rim WP2011 2 0.135 0.0956 -5.60 0.0170 0.0120
## 3 3A-097 WP2011 3 0.0273 0.0158 -6.75 0.160 0.0924
## 4 3B-021 WP2011 3 0.0394 0.0227 -3.68 0.401 0.231
## 5 3E-001clean WP2011 3 0.0516 0.0298 -5.92 0.297 0.171
## 6 3F-019 WP2011 1 Inf Inf -6.95 Inf Inf
## 7 3H-014 WP2011 4 0.0801 0.0400 -5.94 0.168 0.0839
## 8 4-038 WP2011 3 0.0510 0.0294 -7.23 0.136 0.0784
## 9 6-003 WP2011 2 0.0182 0.0129 -6.85 0.00613 0.00434
## 10 6-042 WP2011 1 NaN NaN -6.27 NaN NaN
## # ... with 14 more rows, and 28 more variables: d180.avg <dbl>, D47.sd <dbl>,
## # D47.se <dbl>, D47.2se <dbl>, D47.90.noAD.avg <dbl>, D47.90.avg <dbl>,
## # D47.25D.avg <dbl>, D47.25P.avg <dbl>, D48.sd <dbl>, D48.se <dbl>,
## # D48.2se <dbl>, D48.90.avg <dbl>, D48.sd.all <dbl>, D48.se.all <dbl>,
## # D48.2se.all <dbl>, D48.90.avg.all <dbl>, T.B <dbl>, T.B.se <dbl>,
## # T.G <dbl>, T.G.se <dbl>, T.P <dbl>, T.P.se <dbl>, d180w.B <dbl>,
## # d180w.B.se <dbl>, d180w.G <dbl>, d180w.G.se <dbl>, d180w.P <dbl>,
## # d180w.P.se <dbl>
```

same plot of D47 v D48, but plotting all the individual values, instead of the averages

```
pD47vD48.all <- ggplot(data = data.frame(x=c(0.1, 0.7)), aes(x)) +
  stat_function(fun = function(x) {0.1829*(x^2) + 0.2138*(x) - 0.0139}, colour = "light gray") + #D48 of C02, from
  # Fiebig eq. for Wang et al., 2004
  stat_function(fun = function(x) {0.3821*(x^2) + 0.0044*(x) + 0.1198}) + #from data in Hill et al 2014; regressi
  # on from values in Table B2
  geom_point(data = data, aes(x=CDES.noAD.pri, y=CDES.48noAD.pri, fill = Sample.ID, colour = NULL, shape = D48exc
  # ess.label), size = 2) +
  scale_shape_manual(values = c(21,23,24)) +
  theme_bw()

ggplotly(pD47vD48.all)
```



```
write.csv(data, file=paste0(session, "_brandcorr_reprocessed.csv"))
```

Can subdivide the dataset into individual projects, each with all of the "all" data, to use for supplemental files for different projects. To use, you'll need to manually update the separate project dataframe names and filtering criteria, and then the worksheet labels accordingly

```
Supp.table <- data %>%
  filter(Project == "Marina" | Project == "all") %>%
  select(Date, Sample.ID, Type, Donotuse, runinfo, spec.num, mass, num.acq, d45,d45.stdev,d45.sterr,d46,d46.stdev,d47,d47.stdev,d47.sterr,d48,d48.stdev,d48.sterr,d49,d49.stdev,d47,d47.stdev,d47.sterr,d48,d48.stdev,d48.sterr,d13C,d13C.stdev,d13C.sterr,d180.VPDB.min,d180.stdev,d180.sterr,d48.HGcorr,D48excess.label, HGcorr, D47.HG.prop.err,CDES.noAD.pri,CDES.D47stdcorr90.pri, CDES.ADP.pri,CDES.D47stdcorrP.pri,CDES.ghosh.temp.pri.D,CDES.peterson.temp.stdcorr.pri,CDES.Bonifacie.temp.pri)

samples <- data %>%
  filter(Project == "Marina") %>%
  select(Date, Sample.ID, Type, Donotuse, runinfo, spec.num, mass, num.acq, d45,d45.stdev,d45.sterr,d46,d46.stdev,d46.sterr,d47,d47.stdev,d47.sterr,d48,d48.stdev,d48.sterr,d49,d49.stdev,d47,d47.stdev,d47.sterr,d48,d48.stdev,d48.sterr,d13C,d13C.stdev,d13C.sterr,d180.VPDB.min,d180.stdev,d180.sterr,d48.HGcorr,D48excess.label, HGcorr, D47.HG.prop.err,CDES.noAD.pri,CDES.D47stdcorr90.pri, CDES.ADP.pri,CDES.D47stdcorrP.pri,CDES.ghosh.temp.pri.D,CDES.peterson.temp.stdcorr.pri,CDES.Bonifacie.temp.pri)

Averages <- data.avg %>%
  filter(Project == "all" | Project == "Marina" | Project == "int.standards")

add_ws_with_data <- function(wb, sheet, data) {
  addWorksheet(wb, sheet)
  writeData(wb, sheet=sheet, data)
  return(wb)
}

wb <- createWorkbook("data")
wb <- add_ws_with_data(wb, "all data", data)
wb <- add_ws_with_data(wb, "all supp info", Supp.table)
wb <- add_ws_with_data(wb, "all replicates", samples)
wb <- add_ws_with_data(wb, "Averages", Averages)
saveWorkbook(wb, paste0(session, "_flatlist_project.xlsx"), overwrite = TRUE)
```